IMAGE CODE ESTIMATION

CROSS-REFERENCE TO RELATED APPLICATIONS

The following applications disclose related subject matter: Appl. Nos. 10/...., filed .... These referenced applications have a common assignee with the present application.

BACKGROUND OF THE INVENTION

The present invention relates to digital image processing, and more particularly to compressive encoding of images.

In many applications of digital image processing, such as JPEG, MPEG or DV, image data is compressed by successive operations of DCT (Discrete Cosine Transform), quantization, and Huffman (variable length) encoding. DCT is a pre-processing operation for image compression which converts blocks of spatial domain information of the image into blocks of frequency domain information, typically for 8x8 blocks of pixels. Generally, the transformed image has a tendency to strong correlations within a neighborhood; DCT processing concentrates a majority of the information of the image into the low frequencies. Quantization is an effective compression method which divides the DCT coefficients by an integer quantization level so as to reduce the precision (number of significant bits) of the DCT coefficients. The quantized DCT coefficients of a block are scanned from low frequency to high frequency and converted into a sequence of pairs of 'run' and 'level' parameters (run length encoding). Using a Huffman code table defined by the statistics of an image, the sequence of run-level pairs are finally converted to a sequence of Huffman (variable length) codewords. The Huffman tables for differing sets of variables of JPEG, MPEG, or DV are fixed in the specifications of these standards.

In these image compression methods, the size of the Huffman code generated is strongly dependent on the quantization level. Therefore, it is necessary to apply a suitable quantization level to adjust an output code size to a target code size. One useful quantization level approach has a feedback loop

comparing a target code size and a code size actually generated by the encoder circuit. This approach enables estimation and response to a current code size accurately because actual output code size is used as a comparison to the target code size. However, there is a delay corresponding to the speed of the encoding algorithm to output the actual code. As a result, there is a delay for convergence to the target code size.

SUMMARY OF THE INVENTION

The present invention provides estimation of the code size for digital image compression by simplifying the feedback flow for the quantization process for DCT coefficients and Huffman encoding.

This has advantages including more efficient quantization without time delay of actual code generation.

BRIEF DESCRIPTION OF THE DRAWINGS

The drawings are heuristic for clarity.

Figures 1a-1b show functional block diagrams of preferred embodiment method and preferred embodiment system.

Figure 2 illustrates partitioning a block.

Figure 3 shows an approximation function.

Figure 4 depicts experimental results.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

1. Overview

Preferred embodiment image processing methods include low complexity estimates of the code size for Huffman (variable length code) encoding a block of quantized DCT coefficients; this provides quantization level feedback information for selection of a quantization level(s). The method uses a histogram of the non-zero DCT coefficient magnitudes of (an area of) a block together with normalized code-size functions (a function for representative "level"s and depending upon an average "run") from the Huffman table. The average "run" in the code is estimated from the number of zero and non-zero quantized coefficients in (the area of) the block. Figure 1a is a functional block diagram for a preferred embodiment method.

Preferred embodiment digital image systems (such as cameras) include preferred embodiment image processing methods. Figure 1b shows in functional block form a system (digital stiall camera) which incorporates preferred embodiment methods as illustrated in the JPEG compression block. The functions of preferred embodiment systems can be performed with digital signal processors (DSPs) or general purpose programmable processors or application specific circuitry or systems on a chip such as both a DSP and RISC processor on the same chip with the RISC processor as controller. Further specialized accelerators, such as CFA color interpolation and JPEG encoding, could be added to a chip with a DSP and a RISC processor. Captured images could be stored in memory either prior to or after image pipeline processing. The image pipeline functions could be a stored program in an onboard or external ROM, flash EEPROM, or ferroelectric RAM for any programmable processors.

2. First preferred embodiment

Figure 1a is a functional block diagram of a first preferred embodiment method of quantization level determination and which includes the following steps. First prepare a histogram of the magnitudes of the DCT coefficients of a

(8x8) block; this enables easy estimation of the histogram resulting from a quantization process and reveals the magnitude tendencies which impact the actually generated code size. For a fixed-point processor, a definition of the histogram bins as the ranges $2^n \sim 2^{n+1}-1$ ($n=0,1,2,\ldots$) is especially desirable for the simplicity of implementation. Note that 0 coefficients do not appear in the histogram. Also, quantization amounts to integer division by the quantization level, so quantization levels as powers of 2 permit binary shifting for the division.

Many of the specifications in standards such as JPEG, MPEG, take the DCT coefficient quantization level to depend upon the DCT frequency; that is, a non-flat quantization matrix. Such frequency-dependent quantization levels use the fact that the sensitivity of the human visual system to high spatial frequency components in an image is less than that to low spatial frequency components. In this case, partitioning the DCT coefficient block into several areas which correspond to spatial frequency ranges increases the accuracy of the estimation of generated code size. Figure 2 shows an example of this partitioning into four areas and is applicable to the DV standard encoding process. Note the zig-zag scan order of the DCT coefficients aligns to this partitioning so that the areas are sequentially scanned. Thus, if each DCT block is partitioned into j areas and the magnitudes of the DCT coefficients can be represented by N bits and the histogram bins are taken to be the ranges $2^n \sim 2^{n+1}-1$ ($n=0,1,2,\ldots, N-1$), define the following histogram variable:

$H(i,j)$: the number of coefficients in bin $2^i \sim 2^{i+1}-1$ ($i = 0,1,2\ldots, N-1$) in area j

The preferred embodiment methods use such a histogram with bins of the ranges $2^n \sim 2^{n+1}-1$ for a fast approximation of the quantization process. In particular, for the case of the quantization levels defined in a specification under consideration confined to 2^M ($M=0,1,2\ldots$), the quantization level changes are achieved by shifting values of each bin of the histogram. That is, with quantization level 2^M the number of quantized coefficients, $Hq(i,j)$, within the bin $2^i \sim 2^{i+1}-1$ ($i=0,1,2\ldots, N-1$) in area j is given by:

$$Hq(i,j) = \begin{cases} H(i+M,j) & (0 \le i \le N-M-1) \\ 0 & (others) \end{cases}$$

Even if the quantization levels defined in the specification under consideration are arbitrary integers, the histogram after the quantization $Hq(i,j)$ can be represented approximately by scaling a graph of the distribution given by the original histogram $H(i,j)$.

The variable-length code size of the block after quantization is estimated from the given $Hq(i,j)$ by simplifying the Huffman table. Inputs for the Huffman table are the parameters "run" and "level", so consider an expressions for these parameters using $Hq(i,j)$. Note that the sign of a coefficient typically is the last bit of the codeword, so coefficient differing by sign have the same magnitude and same codeword size. The parameter "run" is the number of zero coefficients between significant coefficients when scanning the DCT block from the low frequency (upper lefthand portion in Figure 2) to the high frequency (lower righthand portion), the so-called zig-zag scanning. If the total number of coefficients in area j is denoted $A(j)$, the parameter "run" in area j can be estimated as the number of 0 coefficients divided by the number of non-zero coefficients. Now the number of non-zero coefficients in area j equals $\sum_i Hq(i,j)$, so estimate "run" by $r'(j)$ as follows.

$$r'(j) \equiv \frac{A(j) - \sum_i Hq(i,j)}{\sum_i Hq(i,j)}$$

As to the parameter "level", it already is implicit in the histogram bins for the magnitudes of the DCT coefficients in area j. Therefore, if an average value is utilized as a representative for the bin of the range $2^i \sim 2^{i+1}-1$, the parameter "level" can be estimated for coefficients in the bin by $l'(i)$:

$$l'(i) \equiv \frac{2^{i+1} - 1 + 2^i}{2}$$

This can also be simplified to the maximum value $2^{i+1}-1$ in the case that the specification under consideration has a restrictive upper limit on the generated code size. That is,

$$l'(i) \equiv 2^{i+1} - 1$$

In any case, the representative value $l'(i)$ is not dependent on the area j. Now, $r'(j)$ and $l'(i)$ provide input parameters for the Huffman table. If the code size of the entry "run" = r and "level" = l in the Huffman table under consideration is denoted as $T(r, l)$, then the code size $c(j)$ for the coefficients in area j of the DCT block can be estimated as follows.

$$c(j) = \sum_{i=0}^{N-1} Hq(i,j)T(r'(j),l'(i))$$

In this equation, $T(r, l)$ is only applied to the representative value, $l'(i)$, for each bin i. Therefore, for this estimation, an original Huffman table can be reduced to the table restricted to the representative "level"s. However, this calculation contains an undesirable division operation from the definition of $r'(j)$. The division can be avoided by using a normalized function $T'(x,i)$ for each bin i where x is a normalized variable of the number of non-zero coefficients applicable to all areas. As shown in the expression for $r'(j)$, it is determined by the fixed number of coefficients in area j, $A(j)$, and the variable number of non-zero coefficients in area j, $\Sigma Hq(i,j)$. By normalizing (scaling) the number $A(j)$ to a convenient integer A $(=\alpha(j)A(j))$ and the number $\Sigma Hq(i,j)$ to $\Sigma Hq(i)$ $(=\alpha(j)\Sigma Hq(i,j))$, a function of the Huffman table $T(r'(j), l'(i))$ can be re-defined independent of area j. Thus, code size $c(j)$ can be represented as follows :

$$c(j) = \sum_{i=0}^{N-1} Hq(i,j)T'(x,i)$$

$$x = \alpha(j)\sum_{j} Hq(i,j) \quad \left(\alpha(j) = \frac{A}{A(j)} \right)$$

That is, x is the number of non-zero coefficients in $A(j)$ normalized to a total of A

$$l'(i) \equiv \frac{2^{i+1} - 1 + 2^i}{2}$$

This can also be simplified to the maximum value $2^{i+1}-1$ in the case that the specification under consideration has a restrictive upper limit on the generated code size. That is,

$$l'(i) \equiv 2^{i+1} - 1$$

In any case, the representative value $l'(i)$ is not dependent on the area j. Now, $r'(j)$ and $l'(i)$ provide input parameters for the Huffman table. If the code size of the entry "run" = r and "level" = l in the Huffman table under consideration is denoted as $T(r, l)$, then the code size $c(j)$ for the coefficients in area j of the DCT block can be estimated as follows.

$$c(j) = \sum_{i=0}^{N-1} Hq(i,j) T(r'(j), l'(i))$$

In this equation, $T(r, l)$ is only applied to the representative value, $l'(i)$, for each bin i. Therefore, for this estimation, an original Huffman table can be reduced to the table restricted to the representative "level"s. However, this calculation contains an undesirable division operation from the definition of $r'(j)$. The division can be avoided by using a normalized function $T'(x,i)$ for each bin i where x is a normalized variable of the number of non-zero coefficients applicable to all areas. As shown in the expression for $r'(j)$, it is determined by the fixed number of coefficients in area j, $A(j)$, and the variable number of non-zero coefficients in area j, $\Sigma Hq(i,j)$. By normalizing (scaling) the number $A(j)$ to a convenient integer A ($=\alpha(j)A(j)$) and the number $\Sigma Hq(i,j)$ to $\Sigma Hq(i)$ ($=\alpha(j)\Sigma Hq(i,j)$), a function of the Huffman table $T(r'(j), l'(i))$ can be re-defined independent of area j. Thus, code size $c(j)$ can be represented as follows :

$$c(j) = \sum_{i=0}^{N-1} Hq(i,j) T'(x,i)$$

$$x = \alpha(j) \sum_j Hq(i,j) \quad \left(\alpha(j) = \frac{A}{A(j)} \right)$$

That is, x is the number of non-zero coefficients in $A(j)$ normalized to a total of A

coefficients; hence, the average "run" in $A(j)$ is $r'(j) = (A - x) / x$ and so the definition $T'(x,i) = T((A - x) / x, l'(i))$ essentially precomputes the division in $r'(j)$. Figure 3 shows an example of expressions for $T'(x,i)$ for $i=0,1,\ldots,7$ which are applicable to the Huffman table in the DV specification. In Figure 3, the number of coefficients is normalized to $A=128$. It is necessary to define $T'(x,i)$ only for $i=0,1,2,3,4$ because the defined table for "level" over 32 has the corresponding size of the Huffman code depending only on "run".

As a result, a total code size, S, for the DCT block can be approximated as

$$S = \sum_j c(j)$$

Thus, a total code size for each block of DCT coefficients can be calculated by the summation of the code sizes of the coefficients in the areas in the DCT block which, in turn, is estimated by the histogram for each area by simplified operation for the quantization level and the Huffman coding $(T'(x,i))$ without performing an actual encoding.

As illustrated in Figure 1a, the code size estimation for a tentative quantization level is then used to select a quantization level for subsequent variable-length encoding.

3. Experimental results

Figure 4 shows an result of the preferred embodiment code size estimation method applied to the DV image encoding process using the 8x8 block partitioning shown in Figure 2 and with the normalized code size approximations shown in Figure 3. Most of the estimation (code sizes estimated for 1350 macroblocks) has an accuracy ranged from 60% to 100%. On average, the accuracy of this estimation method is 87%.

4. Modifications

The preferred embodiment code size estimation methods can be varied in various ways while preserving the feature of estimating the code size from a histogram of coefficient magnitudes together with the normalized code size functions from the Huffman (variable length code) table.

For example, the DCT coefficient block size could be increased (e.g., 16x16) or decreased, the number of areas in a block could be varied from 1 to any convenient number, normalization size A for the normalized code size functions $T'(x,i)$ can all be varied, and so forth. The same code size estimation as a function of quantization level could be applied to coefficients of a wavelet transform in place of the DCT.